StratusLab Cloud Distribution

Charles Loomis, Mohammed Airaj, Marc-Elian Bégin, Evangelos Floros, Stuart Kenny, David O'Callaghan

StratusLab Cloud Distribution

Charles Loomis[1] Mohammed Airaj[1,2] Marc-Elian Bégin[3]
Evangelos Floros[4] Stuart Kenny[5]
David O'Callaghan[5]

[1]Laboratoire de l'Accélérateur Linéaire, Orsay, France
[2]Université Caddi Ayyad, Faculté des Sciences et Techniques, Marrakech, Morocco
[3]SixSq Sàrl, Geneva, Switzerland
[4]Greek Research and Technology Network S.A. (GRNET), Athens, Greece
[5]Trinity College Dublin, Dublin, Ireland

This work originally appeared as Chapter 10 in
European Research Activities in Cloud Computing
Dana Petcu and Jose Luis Vasquez Poletti (editors)
Cambridge Scholars Publishing (Jan. 2012)
ISBN: 1-4438-3507-2; ISBN13: 978-1-4438-3507-7

Published with permission of Cambridge Scholars Publishing

Abstract

Cloud technologies provide many benefits for scientific and engineering applications, such as customised execution environments, near-instantaneous provisioning, elasticity, and the ability to run user-level services. However, a rapid, wholesale shift to using public, commercial cloud services is unlikely because of capital investments in existing resources and data management issues. To take full advantage of cloud technologies in the short term, institutes and companies must be able to deploy their own cloud infrastructures. The StratusLab project provides a complete, open-source cloud distribution that permits them to do this. The StratusLab services include the computing, storage, and networking services required for an Infrastructure as a Service (IaaS) cloud. It also includes high-level services like the Marketplace that facilitates the sharing of machine images and Claudia that allows the deployment and management of complete software systems.

1 Introduction

Despite the enormous marketing hype around cloud technologies, much of the excitement is well deserved. Cloud infrastructures provide users with more agility by avoiding lengthy procurement delays, access to larger computing resources by avoiding constraints from local physical infrastructures, customised execution environments, and the possibility to expose user-level services rather than just static content.

Cloud computing has been embraced by small and medium-sized enterprises (SME) as a means to limit capital expenditures and to allow for fast, even exponential growth. Those in the academic sector have also taken notice and are investigating how cloud computing can provide results more quickly and at lower cost. This is especially true for academic areas that traditionally lack experience with deploying and maintaining local computing infrastructures, like the humanities and social sciences.

The desire to maximise elasticity at minimum cost pushes public cloud providers to deploy massive data centres in locations where energy and cooling costs are low. Environmental concerns force providers to consider also the carbon footprint of their operations. Computing infrastructures are swinging back to the centralised model of the mainframe era.

However, several factors slow this movement towards a centralised computing model based on public cloud infrastructures. Inertia is one factor, where existing investments in physical (and human) infrastructure mean that current resources cannot be immediately outsourced to a cloud. User confidence in the cloud is another. Users must be convinced that cloud provides a reliable, robust service, especially in the face of recent large-scale outages of Amazons cloud [19].

Data management, however, is the most fundamental impediment. On the commercial side, privacy, confidentiality, and legal constraints hinder remote storage of data, although some commercial offerings are now appearing to address this issue. On the academic side, data are generated by a variety of instruments, are naturally distributed, and need co-located computing resources. Because of this, there will be a need for local computing resources at companies and at universities for some time to come.

Companies and universities still want to take advantage of the numerous benefits of cloud technologies. To do so in the short term, they must be able to deploy a cloud in addition to or instead of using a commercial cloud provider. The StratusLab project offers a complete, open-source, Infrastructure-as-a-Service (IaaS) cloud distribution to satisfy this need.

2 StratusLab Project

The StratusLab project [26] funded in part by the European Commission, aims to develop a complete, open-source cloud distribution that can be deployed in production in both the academic and commercial sectors. It consists of six partners from five European countries (France, Greece, Ireland, Spain, and Switzerland) and runs for two years, ending in May 2012. All of the partners have previous experience with virtualisation and with distributed computing, particularly grid computing.

2.1 Cloud Technology

The term "cloud," as used in the press and popular literature, describes a wide range of technologies and deployment strategies. Marketing deliberately misuses the term to take advantage of the current excitement surrounding the technology, and further muddies the water in doing so. NIST, fortunately, has published excellent definitions of the different types of cloud service and deployment models [18]. Table 1 defines the different cloud service models, based on this publication. The cloud service models form a rough hierarchy of services, where platforms are built on IaaS clouds and services are built on PaaS clouds.

There are also a variety of deployment models. Private clouds are run by an organisation for its own users whereas public clouds are run by a provider for the general public. These have significantly different security requirements although the underlying cloud technologies are the same. A community cloud is an intermediate deployment between private and public clouds providing a collaborative infrastructure involving users and possibly providers at different institutes. Hybrid cloud infrastructures bridge different cloud deployments to federate the available resources.

The underlying ideas of clouds are not new, having their roots in the commodity and utility computing of the past two decades. Those initiatives, however, had limited success compared to cloud computing. The popularity of cloud computing comes from the convergence of three factors:

- Mature virtualization technologies imposing very small performance penalties compared to bare-metal performance,
- Simplified APIs and usage models, particularly RESTful [22] APIs running over the standard HTTP(S) protocol, and
- Excess commercial computing capacity.

This last point is especially important as it allowed cloud computing to be made available to a wider audience at a reasonable price.

The StratusLab cloud distribution allows an IaaS-type cloud to be deployed; it also provides functionality to support PaaS cloud installations over the IaaS cloud. The distribution supports any type of cloud deployment: private, community, or

Table 1: Cloud Service Models

Software as a Service (SaaS)	Consumers use the provider's applications that run on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface.
Platform as a Service (PaaS)	Consumers deploy onto the cloud infrastructure (created or acquired) applications developed using programming languages and tools supported by the provider.
Infrastructure as a Service (IaaS)	Consumers provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications.

public clouds. Future versions will support federation of cloud infrastructures into hybrid clouds.

2.2 Existing Software

Many commercial software vendors now provide solutions for deploying private clouds. Probably the best known is the ESXi suite from VMware. Most academic institutes and smaller companies cannot afford these solutions because the costs can equal or exceed the costs for the hardware itself. They also tend to be less adaptable as users do not have access to the code to make changes for their environment or to experiment with alternate scheduling/provisioning algorithms.

At the time the StratusLab project was proposed, there were no complete, open source cloud solutions available. However, there were many toolkits for virtual machine management. The best-known toolkits are Nimbus [14], Eucalyptus [12] and OpenNebula [8]. Since that time, all of these toolkits have become more functional and provide a larger set of the services necessary for creating an IaaS cloud.

OpenStack [17], another initiative started concurrently with Stratuslab, also aims to provide an open source IaaS cloud distribution. The planned services of OpenStack mirror closely the compute, storage, and networking services in the StratusLab distribution, but are accessible via different APIs. Future evolutions of both platforms will hopefully move toward standard APIs and provide some level of interoperability for users of the two systems.

Because of the desire to provide a production quality cloud distribution quickly, one of StratusLabs core principles is to reuse and build upon existing software when possible, for example OpenNebula and Claudia. The project choose to use OpenNebula for virtual machine management because of its maturity, open architecture permitting extensions, and use of a software license (Apache 2) compatible with commercial reuse of the StratusLab results. Claudia permits the deployment and control of complex, multi-machine services, and consequently is a critical component for supporting Platform as a Service (PaaS) infrastructures above StratusLab clouds. Both of these packages emerged from the RESERVOIR [21] research project and have since been extended and hardened for use within StratusLab.

3 Existing e-Infrastructures

European academics have access to a wide variety of existing infrastructures, ranging from local clusters, through university computing centres, to large European computing platforms. An important goal of the project is to ensure that academics at all levels can take advantage of cloud computing through the StratusLab cloud distribution, while maintaining its relevance also in the commercial sector.

3.1 Commercial

Cloud technology really was born in the commercial sector, stemming initially from Amazons efforts to monetise their excess computing capacity through an IaaS offering. Google with Google App Engine and Microsoft with Azure have since also provided public cloud infrastructures at the PaaS level.

All of these platforms have been popular with start-ups and small enterprises, allowing them to avoid capital expenditures on computing hardware and offering the possibility of quick expansion as the business grows. They have had more limited success for larger enterprises and for academic users, mainly because of data management issues. For commercial entities, issues concern privacy and confidentiality of sensitive data. For academics, the issues revolve around the accessibility of data and its large size.

An open source cloud distribution makes cloud technologies more accessible to larger enterprises by allowing private cloud deployments that overcome the data protection issues. Similarly, cloud deployments (either public or private) near large data stores ensure fast, reliable access to academic data.

3.2 PRACE

The Partnership for Advanced Computing in Europe (PRACE) [4] is an initiative to provide a world-class supercomputing service for European research and industrial users. The service will target users with parallel applications that need very high CPU performance and inter-process communication bandwidth.

These highly optimised applications are unlikely to benefit from cloud computing because of the slight degradation in performance due to virtualisation and because most cloud infrastructures cannot provide high-performance network interconnects. Consequently, this supercomputing infrastructure will remain an important complement to other computing platforms, such as cloud and grid platforms, based on off-the-shelf hardware.

3.3 EGI

The European Grid Infrastructure (EGI) [1] provides a high-throughput computing platform with resources distributed throughout Europe. This platform is open to European researchers or those researchers associated with a European project. The infrastructure itself comprises more than 300,000 CPU cores and 100 PB of disk space, contributed by approximately 350 participating computing centres in 57 countries.

The infrastructure has been very successful in supporting scientific endeavors with over 13,000 users taking advantage of the infrastructures resources. Although open to all scientific disciplines, high-energy physicists, specifically researchers associated with the Large Hadron Collider at CERN, are by far the largest contributor and consumer of resources available through the infrastructure.

For historical reasons, the primary grid middleware used on the infrastructure is gLite [13], although there are on-going efforts to support also UNICORE [5], ARC [15], and Globus [3]. All use a model based on job submission and appear, crudely, as distributed batch systems. This model works well for applications or calculations that can be easily parallelised into large numbers of independent tasks, like those in high-energy physics.

However, many other applications need customised execution environments or involve the deployment of user-level services and are not well adapted to the computing model of the grid. StratusLab, working with EGI, aims to bring cloud technology to this platform to make the infrastructure more flexible and, hence, more attractive to a wider variety of users.

4 Combining Grid and Cloud Technologies

Some view grid and cloud technologies as competing technologies; this project however views them as complementary. Combining those technologies promotes scientific e-infrastructures that are more dynamic, appeal to a wider variety of scientific communities, and encourage international and interdisciplinary collaboration.

Models to combine these technologies can be crudely categorised as “grid-over-cloud” or “cloud-over-grid.” The cloud-over-grid approach uses the existing grid job submission and management services to launch (customised) images for job execution. The CNAF data centre in Italy has used this approach for their WNoDeS [6] service. This does provide a customised environment for users, but still presents new users with the high learning curve of grid services and still maintains a “job-based” workflow that is not applicable to all applications.

StratusLab takes the opposite approach–“grid-over-cloud”–specifically treating the grid as a Platform-as-a-Service (PaaS) infrastructure running over StratusLabs Infrastructure-as-a-Service

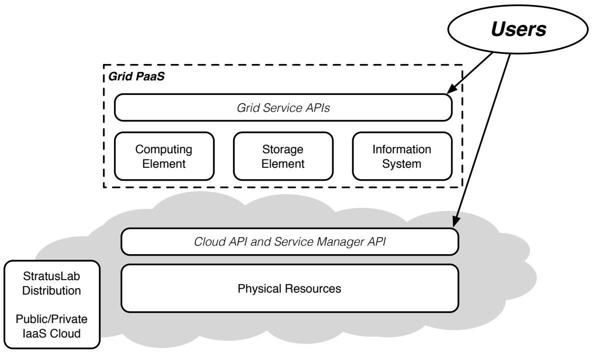

Figure 1: Grid as a PaaS

(IaaS) cloud (see Figure 1). This architecture allows system administrators to expose their resources via both grid and cloud interfaces to their users, maximising the sites utility. By offering both interfaces, it allows users to decide which abstraction is best suited to their applications: a job-based abstraction for the grid or a machine-based abstraction for the cloud. Users can exploit fully the dynamic nature of the cloud, provisioning resources as needed and running user-level (and community-level) services as appropriate.

However, the layering of grid services over the cloud does require some detailed integration to ensure that they interoperate well, especially from the users point-of-view. For example, the grid and cloud interfaces should support the same authentication and authorisation mechanisms to ensure that users can take advantage of both types of services and potentially mix them for a particular calculation.

To test this scenario and ensure that it works, StratusLab runs a production grid site within EGI. This site has been certified via the standard EGI procedures and is monitored and evaluated with the standard tools. This experience has shown that StratusLab can indeed support running grid services within a cloud, although direct use by scientists at this point has been limited. With the production release of the StratusLab distribution, the project expects pilot sites to adopt this architecture, providing further information on running this scenario and allowing the project to further refine the distribution.

5 Requirements

At a minimum, the StratusLab cloud distribution must provide all of the services for a complete IaaS cloud deployment, including compute, storage, and networking services. The compute services must allow the deployment of virtual machines created by users themselves and should pose no constraints on the type of operating system used within the virtual machine. Similarly, a standard, lightweight contextualisation method must be used to minimise perturbations to the users standard machine configuration. The storage services must allow persistent storage of data with a lifecycle independent from a particular virtual machine instance. The system must allow services running on virtual machine instances to be accessible from the wide area network, if requested by the user.

The StratusLab distribution must also support operation in a distributed, collaborative environment like that of EGI. This requires a flexible authentication framework that allows integration with grid authentication mechanisms as well as existing authentication mechanisms at sites deploying standalone cloud infrastructures. The distribution must also be designed to foster collaboration by, for instance, allowing the easy sharing of virtual machine images. Many scientific domains will build their own computing platforms above an IaaS cloud; consequently, the distribution must provide services to facilitate the deployment and operation of these domain-specific PaaS.

StratusLab targets a wide range of data centres, ranging in size from a few tens of machines to tens of thousands. Consequently, the services must be scalable. The experience of the system administrators at the targeted sites also varies greatly. As a result, the distribution should be easy to install but also be compatible with the automated installation tools of larger sites. Finally, different sites prefer different operating systems, so the distribution should demonstrate portability between several operating systems.

6 Distribution

The project uses agile software development methodologies to ensure that the StratusLab cloud distribution meets the requirements of our users

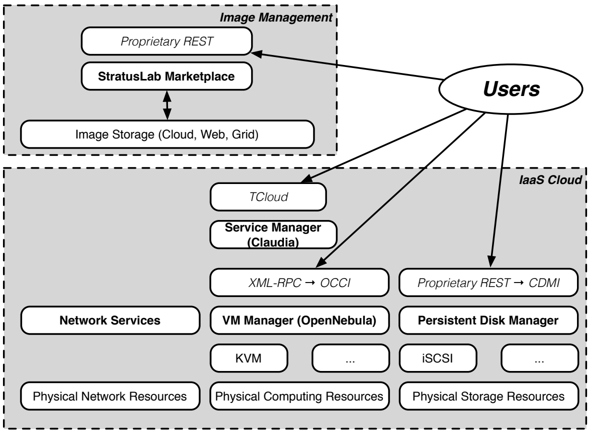

Figure 2: StratusLab Architecture

and evolves quickly while maintaining its production quality. The project always moves from working version to working version, making iterative improvements and testing continuously, ensuring that all versions can potentially be released.

Because of this methodology, StratusLab released at the end of its first year the 1.0 production distribution, preceded by four public beta releases. The first beta release, three months into the project, was already a functional cloud; the project has been running a reference cloud infrastructure with its software since the second beta release.

A StratusLab cloud based on the 1.0 production release offers computing, storage, and networking services as well as high-level services to facilitate sharing of machine images and deployment of multi-node services (see Figure 2). For users, the distribution also provides a set of client commands for interacting with the cloud services. These commands are written in python for portability, running on Linux variants, Mac OS X, and Windows. For system administrators, deployment can be done via a scripted manual installation or via a fully automated configuration with the Quattor [20] fabric management system.

6.1 Computing

Virtual machines are at the centre of any IaaS cloud, and consequently virtual machine management is a core feature of any IaaS cloud distribution.

The StratusLab software stack for computing services consists of the operating system running on the physical hardware, a hypervisor for local VM control, a virtual machine manager (VMM) for cloud-wide management of VMs, and a cloud API permitting remote access to the resources. Currently, StratusLab supports Fedora14 at the OS-level, uses KVM for the hypervisor, integrates OpenNebula as the VMM, and exposes OpenNebulas proprietary XML-RPC interface as a cloud API.

OpenNebula allows users to control the complete lifecycle of a virtual machine from its initial creation to ultimate destruction. It provides many hooks into the machine lifecycle and various customisation points that allow it to integrate well with the other StratusLab services.

The project has also worked with the developers to improve the system. To help with forensic analysis following security incidents, a quarantine system was introduced and more detailed logging is provided. Other improvements include better management of network addresses, better error feedback to users of the cloud API, and eventually user, group, and role support in the authorisation system.

Future versions of StratusLab will take better advantage of the flexibility of OpenNebula, allowing the support of more underlying operating systems and more hypervisors. The project will also move to OCCI [16], a proposed OGF standard, for the cloud API, potentially allowing users a common API for different cloud implementations.

6.2 Storage

All non-trivial analyses use input data for a calculation and store its result. Consequently, storage resources are a core component of any computing infrastructure. StratusLab uses a disk-based abstraction for its storage services providing tools to manage and to exploit both static (read-only) disks and persistent (read-write) disks.

A static disk is essentially a file containing a disk image that can be mounted on one or more virtual machines. As implied by the name, these disks are immutable and only provide read-only access to the data when mounted on virtual machines. These disks are useful for defining fixed databases for use (and reuse) for scientific calculations.

Because these are read-only files, they can be treated in the same way as virtual machine images.

Consequently, the Marketplace (see below) can be used as a catalogue of available static disk images and the standard StratusLab caching mechanisms are used to minimise transfers and access latencies to these data.

StratusLab provides a separate service for persistent disks. These disks have a lifecycle–creation, use, and removal–independent of the virtual machines that use them. The persistent disk service currently sports a proprietary RESTful interface that allows control of a disks lifecycle. The service can use either an iSCSI server (with disk storage behind) or a standard shared file system (e.g. NFS or GPFS) for the sharing of the persistent disks with the compute nodes.

The service is integrated with the compute services to allow the transparent mounting and dismounting of these disks on running virtual machines, either at boot time or while a machine is running.

The project has put a low priority on file-based storage services for two reasons: (a) most users already have access to file-based storage through grid storage services, WEBDAV-enabled servers, or similar services; and (b) a disk-based abstraction makes more sense when working at the Infrastructure-as-a-Service level that StratusLab targets.

SNIA [24] has proposed Cloud Data Management Interface (CMDI) [23] as a standard for storage services. This is a RESTful API for storage that has a largely file-based abstraction for storage. Treating the virtual disks as files within this API makes sense. Consequently, the persistent disk service will be enhanced to provide a CDMI interface.

6.3 Network

To connect users with distributed resources, cloud computing relies critically on a fast and stable network infrastructure. For the European Research Area, GÉANT [2] provides this infrastructure in partnership with national organizations. Commercial entities, however, cannot use the research network and must instead rely on commercial networking providers.

Advanced networking services, including dynamic Virtual Local Area Networks (VLAN), are not needed for basic use of a cloud (and in fact are not needed at all for private cloud deployments). However, for public cloud deployments, such services can increase the security and help alleviate data protection concerns.

The current releases of the StratusLab Cloud Distribution provide a very basic level of networking services, relying on the existing production network and permitting the allocation of "public," "local," or "private" IPv4 addresses. These three types of addresses correspond to three distinct use cases for the StratusLab cloud.

Public IP addresses are visible from inside and outside of a cloud deployment. They permit services running on a cloud to function as real services with visibility to users outside of the organisation running the cloud infrastructure. As they are public, they also require a higher level of security at the level of the running virtual machine. A disadvantage of public IPv4 addresses is that they are becoming increasingly scarce.

Local IP addresses are visible from virtual machines running within a cloud deployment. Access to external services from the virtual machines requires going through a NAT (Network Address Translation) server. Despite the more limited visibility of these addresses, they can be used to communicate between related machines within a cloud. Examples of this include nodes involved in parallel calculations (for example MPI calculations) and worker nodes in a batch system.

Private IP addresses are visible only from the physical host on which a virtual machine is running. Access to external services takes place through NAT on the physical host. The very limited visibility of these addresses increases the security of the machine. Despite the limited connectivity, machines running with private IP addresses can participate in master/worker frameworks such as BOINC [9] or XtremWeb [7].

Having three different types of IP addresses available allows users to balance the requirements of services running within their virtual machines with potential security improvements due to limited visibility of the services. This also allows scarce public IPv4 addresses to be conserved for those virtual machines that really need to be externally visible.

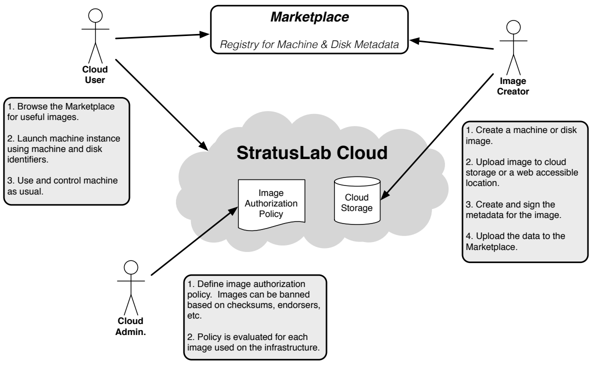

Figure 3: Marketplace Workflows

The above scheme has turned out to be very flexible, permitting a large variety of services to be deployed on a StratusLab cloud. Nonetheless, the distribution will incorporate improvements to further improve the security of running machines: provision of VLANs and dynamic firewalling. Another important topic for the future will be the support of IPv6 addressing, both for the clouds physical hosts and for the virtual machines running within the cloud.

6.4 Marketplace

Creating robust and secure virtual machine images requires significant experience. For many people, the necessity to create a virtual machine image is a significant barrier to the use of cloud resources. Sharing virtual machine images can allow many people to overcome this barrier.

StratusLab provides a "Marketplace" that allows: (a) users to browse for relevant, pre-existing virtual machine images; (b) system administrators to evaluate virtual machine images against their security policies; and (c) machine creators to publish their work to a larger audience (see Figure 3). The Marketplace is a searchable database containing metadata entries for (both disk and machine) images; searches can be done programmatically or via a web browser. The image content itself is hosted in cloud storage or on an external server.

User Workflow Users search for images based on the metadata, then use the image identifier to launch a virtual machine or to attach a static disk to a machine. The metadata contains basic information such as the operating system, version, and architecture, as well as service information such as what applications are available in the image. The required entries are kept to a bare minimum, instead allowing communities to adapt and to extend the metadata as necessary.

System Administrator Workflow System administrators define a policy that validates images according to the sites requirements. All images used are then validated against the policy before being deployed within the cloud. The metadata information is cryptographically signed, allowing the administrator to decide if an image can be trusted based on the endorser and the metadata. Service information such as required ports allow the system to determine if the machine can run correctly within the given cloud deployment.

Creator Workflow Image creators upload their image to cloud storage or make it available via an external service. They then create a metadata entry describing the image, sign the entry, and upload it to the Marketplace. In so doing, they endorse the image. This endorsement can be revoked by uploading another metadata entry that marks the image as deprecated. An interesting point is that people other than the creator can endorse the image, for example marking it as certified by an independent evaluator.

The implementation is very open, allowing the largest number of people to make use of the Marketplace. At the same time, it permits the uploaded information to be validated and evaluated against security criteria. See the Marketplace Technical Note [25] for the complete list of security concerns.

The core functionality of the Marketplace is in place. Changes are likely to come in the content of the metadata entries as the user community discovers what information is necessary to effectively share images. It is also likely that social features will be added to the Marketplace, allowing the reliability of images and endorsers to be determined by the community.

6.5 Claudia

Using an IaaS cloud service is effective for single machine services and for building platforms with multiple cooperating machines.

For platforms however, it often becomes tedious to have to synchronise the deployment of the various machines (for example, servers must be deployed before clients) and to provide manual failover strategies. High-availability and auto-configuring services can reduce the need for manual interventions. Unfortunately, such services are still very much in the minority at most data centres.

Claudia, a “service manager” aims to automate the deployment of an ensemble of machines (a “service”) and to provide limited autoscaling of the service. It uses an OVF [10] description that describes both the dependencies between machines and rules for changing the deployment based on performance or load metrics.

Claudia is a part of the StratusLab production release and has been used to automatically deploy a batch system and a complete grid site. It currently does not use the standard StratusLab authentication mechanism and is thus appropriate only for private cloud deployments.

The service exposes the TCloud [27] interface, which has been submitted to the DMTF [11] as a proposed standard. As the service evolves, the standard authentication mechanisms will be incorporated to allow use on public cloud deployments. Incremental improvements in the functionality and robustness of the code will continue over the lifetime of the project.

7 Reference Infrastructure

Although the project does not intend to provide a perennial e-infrastructure for scientists, it does provide a reference cloud infrastructure to demonstrate the StratusLab cloud distribution, to allow production use of the software, and to encourage feedback on the projects software.

This infrastructure is provided by GRNET in Athens, Greece, consisting of 17 physical servers, 256 CPU cores, 768 GB RAM (48 GB/node), 3.6 TB of disk space, and 10 Gb/s network interconnects. To ensure feedback from a wide variety of people, the infrastructure is open to the public (with account registration).

A second infrastructure will also be available at LAL in Orsay, France to give users access to a second infrastructure and to allow the project to investigate various cloud federation schemes.

8 Use of the StratusLab Distribution

The production release is recent, and real world deployments and use of the StratusLab distribution are just beginning. Nonetheless, there are indications that the cloud services are robust and meet the needs of users.

The operation of the reference infrastructure is the first indication. To demonstrate further the functionality and stability of the StratusLab distribution, GRNET also runs a production grid site (HG-07-StratusLab) within the European Grid Infrastructure on top of the reference cloud infrastructure. This site had to go through all of the standard certification procedures to be a part of EGI and is monitored through its standard operational tools. This site validates the model of deploying grid services as a PaaS over a StratusLab cloud and the functionality of the distribution itself. The services have remained stable for more than six months, even in the face of frequent upgrades of the StratusLab software.

A few bioinformatics researchers and engineers act as a pilot scientific community for the project. They have created customised virtual machine images that allow them to access standard bioinformatics databases and to analyse this information with standard software tools within the domain. A more involved analysis, involving the determination of protein structures based on Nuclear Magnetic Resonance data, will determine if the software can effectively handle complicated workflows and permit easy autoscaling in response to the changing needs of the analysis.

SixSq is also currently installing a StratusLab cloud instance to test whether it can run the standard software environment used by the European Space Agency. The ultimate aim is to use cloud technologies to expand the scope of the testing for their software and to speed the deployment of new releases.

Feedback from all of these installations and tests will undoubtedly improve future releases of the distribution and lead to more deployments of the software.

9 Challenges

The current StratusLab distribution provides a solid, production-quality cloud distribution. Nonetheless, there are some challenges to running a cloud that will require further software development and changes in the way non-commercial computing resources are managed. Some aspects of these challenges are being explored within the project, but full solutions will need research and developments extending beyond the end of the project.

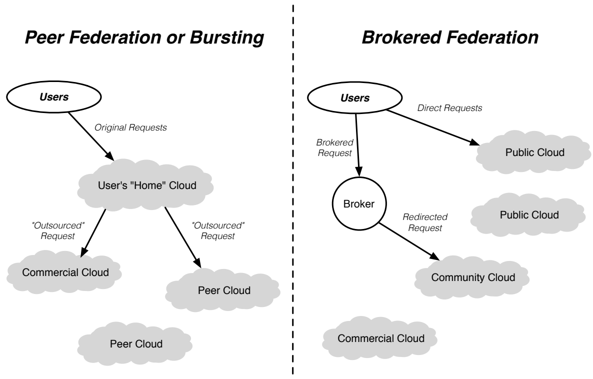

Figure 4: Cloud Federation Models

9.1 Scheduling

Although the usual cloud usage model does not expose queues or scheduling to the end-user, scheduling is critical for cloud providers. On cloud infrastructures, two areas require scheduling decisions: initial placement of virtual machines and optimisation of the total infrastructure.

The initial placement of a virtual machine is very similar to scheduling a job in a traditional batch system. The scheduler must match the requirements of the virtual machine against the (available) capacity of physical machines making up the cloud. All of the accumulated knowledge and experience with batch systems is directly applicable to creating an efficient cloud "placement service."

However, virtualisation technologies also permit virtual machines to migrate from one physical host to another. This opens up new possibilities for adjusting the load on a cloud based on any number of characteristics: energy use, service level agreement management, state of physical hardware, etc. However, migration itself has resource and service availability overheads that must be considered when balancing the load of the system. Developing efficient "optimisation services" for the cloud are critical for efficient management of the resources and driving down the costs of providing them.

9.2 Infinite Capacity

Simplicity and elasticity are the two fundamental characteristics of cloud computing; without them the technology is much less compelling. Both come from a model in which the cloud resource has infinite capacity. In this case, resource requests can be handled immediately, no matter how numerous or large. This simplifies the usage model, avoiding complications like queuing, scheduling, and priorities that are features of other shared computing resources.

No real cloud truly has infinite resources, but for all intents and purposes, large public cloud providers like Amazon or Microsoft have enough to make them appear infinite. Resource centres typical of universities or small companies, however, do not. If these centres are to deploy their own clouds and give their users the same simple usage model, they must maintain the illusion of infinite capacity. How can this be done?

Federation of cloud resources offers one possible solution (see Figure 4). One model has different cloud administrators agreeing to share resources. In this case, requests which cannot be handled by one cloud resource would be outsourced to another peer cloud. This would be completely transparent to the user and maintain the illusion of infinite capacity. This model is usually called "cloud bursting" when commercial public providers are used for the outsourcing.

Another model for federation would expose the individual cloud resources to the user. In this case, if one cloud was at capacity, then the user could choose another cloud resource. No doubt, cloud-brokering services that search for and select particular clouds would quickly spring up, hiding this complexity from the users. This model is very similar to the "brokering" model that is used in current large-scale grid infrastructures, like EGI.

Federation, however, brings its own issues. Many virtual machines work in concert with others or are co-located with the data they process. When outsourcing virtual machines to another cloud, these interdependencies are crucial as network bandwidth is much lower over the wide-area network than within a single cloud resource. In addition, the shipping of data between sites may compromise privacy or confidentiality constraints. Technical solutions to these issues can be found, but these solutions will significantly complicate any federated cloud deployment compared to a single site cloud deployment.

Reintroducing queuing to the cloud usage model offers another partial solution. Not all resource requests need to be handled immediately. Those less urgent tasks could be queued until demand drops and resources become available. The tricks are (a) to identify those tasks that can be queued and (b) to keep the standard (non-queued) usage model simple. Amazon provides a rather elegant solution with its spot instances–virtual machines that run only when the price drops below a certain threshold.

9.3 "Pay as You Go"

All public, commercial clouds have accounting and billing services that allow users to pay for only the resources that they consume. This is a core part of their business model and source of their revenue. This economic model also ensures that users monitor their utilisation and gives the provider control over the total utilisation.

Because cloud providers identify their users through their credit cards and tie utilisation directly to the cards, users are highly motivated to monitor their utilisation, using the most efficient methods for calculations and shutting down unneeded virtual machines. This encourages efficient sharing of the capacity.

Within the economic model, providers have control over the prices. This gives those providers the ability to control the total load of their resources by raising or lowering prices. Essentially this simplifies the load management both for the provider and for the users, by avoiding complicated mechanisms using quotas and fair shares.

Economic models can offer the same benefits in academic settings. The problem is not technological–all public cloud providers deploy and use accounting and billing systems. The problem is overcoming the reluctance to using these models and the inexperience in pricing resources in academic environments. Nonetheless, moving to such a utilisation model (even with "fake" money) would facilitate equitable sharing within an institute and federation between cooperating institutes.

10 Future Directions

Within the first year, the project created a complete, open source cloud distribution, providing the computing, storage, and networking services required for deploying an IaaS cloud. The distribution also includes high-level services like the Marketplace to promote sharing of machine images and Claudia to orchestrate the deployment and control of multiple machine services.

The second year will bring incremental changes to the existing functionality, focusing on improving the ease of use and integration between the services. The services will evolve to support the emerging cloud standards: OCCI, TCloud, CDMI, and OVF. This transition will be transparent for users of the StratusLab command line interfaces.

More profound functional additions to the StratusLab distribution will come from the support for federated cloud infrastructures. This will include cloud "bursting" to commercial public cloud providers or to other StratusLab cloud deployments. It will also include a study on how best to include and to federate cloud resources on the European Grid Infrastructure.

Functional advances are of little benefit without real users of the software. Consequently, a major focus of the second year of the project will be to identify concrete scientific and engineering use cases and to ensure that they can be satisfied using a StratusLab cloud. Already, scientific use cases have been identified in bioinformatics and computational chemistry. An engineering use case involves the deployment and use of the typical software development tools (code versioning system, bug tracker, web site, wiki, LDAP server, etc.) to create a software engineering PaaS.

Longer-range plans include measuring the performance of specific applications on a StratusLab cloud and identifying areas that need improvement. This would allow cloud services to offer High Performance Computing services and expand the range of applications that are well adapted to running in a cloud environment.

11 Summary

Cloud technologies provide many benefits for scientific and engineering applications: customised execution environments, near-instantaneous provisioning, elasticity, and the ability to run user-level services. To provide elasticity while remaining efficient, there is a push towards larger, centralised data centres. However, there are also many barriers that prevent a rapid shift back to centralised computing. On the one hand are the current capital investments and human investments in existing data centres in research institutes and university departments. On the other are concerns about data management, especially the production, availability and maintenance of large scientific data stores, in a centralised model when much of the data is produced in a distributed manner. Because of these barriers, the shift towards a more centralised model based on public cloud infrastructures will be gradual.

If scientists and engineers are to take advantage of the benefits of cloud computing in the short term, their institutes must be able to transform their current resources into a cloud infrastructure. This requires a complete, readily accessible cloud distribution that they can install.

StratusLab has developed and maintains a complete, open source cloud distribution that allows people to install their own public or private IaaS cloud infrastructure. The StratusLab services include the computing, storage, and networking services required for an Infrastructure as a Service (IaaS) cloud. It also includes services like the Marketplace that facilitate sharing of machine images and Claudia that allows the deployment and management of complete software systems.

The project produced four beta releases of the software and its 1.0 production release in the first year of the project. This was possible because it chose to integrate existing solutions where possible (for instance using OpenNebula for the virtual machine management) and to follow an agile software development method that permitted rapid evolution and validation of the StratusLab distribution.

Deploying a cloud infrastructure in the smaller data centres of research institutes and universities does pose some general (not entirely technical) challenges. Elasticity is a key characteristic of cloud computing, but how is this maintained when deployed on finite resources? Federation is a possible answer here, but this can bring complications both for the cloud implementers and users. Virtual machine scheduling policies are extremely important to meet certain goals, like availability, performance, and energy efficiency. How can these be incorporated into cloud deployments? Most commercial providers use a "pay as you go" economic model, both as a means of monetising their infrastructure and as a way to control resource occupancy. How can this be adapted to academic environments that do not have a tendency to use such techniques?

The StratusLab cloud distribution will continue to evolve, incrementally adding new features and improving existing ones. With the initial 1.0 production release as a firm base to build upon, the project will concentrate on supporting hybrid cloud infrastructures and on demonstrating real scientific and engineering use cases. Many of these use cases will eventually become domain-specific platforms running on a StratusLab IaaS cloud infrastructure, allowing European e-Infrastructures to become more diverse while maintaining a common infrastructure to promote sharing and collaboration.

Acknowledgements

StratusLab is co-funded by the European Commission through the 7^{th} Framework Programme (Capacities), contract number INFSO-RI-261552. The authors also acknowledge the support provided by other members of the StratusLab collaboration and by the institutes involved in the project.

References

[1] *European Grid Infrastructure (EGI)*, http://egi.eu/.

[2] *GÉANT*, http://www.geant.net/.

[3] *Globus*, http://www.globus.org/.

[4] *Partnership for Advanced Computing in Europe (PRACE)*, http://www.prace-project.eu/.

[5] *UNICORE*, http://www.unicore.eu/.

[6] *Worker Nodes on Demand Service (WNoDeS)*, http://web.infn.it/wnodes/.

[7] *XtremWeb*, http://www.xtremweb.net/.

[8] B. Sotomayor and R. S. Montero and I. M. Llorente and I. Foster, *Virtual Infrastructure Management in Private and Hybrid Clouds*, IEEE Internet Computing (2009).

[9] Berkeley, University of California, *BOINC*, http://boinc.berkeley.edu/.

[10] Distributed Management Task Force Inc., *Open Virtualization Format Specification*, 22 February 2009, DSP0243.

[11] Distributed Management Task Force, Inc. (DMTF), http://dmtf.org/.

[12] Eucalyptus Systems, *Eucalyptus*, http://www.eucalyptus.com/.

[13] gLite Consortium, *gLite Middleware*, http://glite.org/.

[14] Nimbus Project, *Nimbus*, http://nimbusproject.org/.

[15] Nordugrid, *Advanced Resource Connector (ARC)*, http://www.nordugrid.org/arc/.

[16] Open Grid Forum (OGF) OCCI Working Group, *Open Cloud Computing Interface (OCCI)*, http://occi-wg.org/.

[17] OpenStack, *OpenStack*, http://www.openstack.org/.

[18] Peter Mell and Timothy Grance, *The NIST Definition of Cloud Computing (Draft)*, Tech. report, NIST, January 2011, http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145_cloud-definition.pdf.

[19] Peter Svensson, *What Amazon's Outage Means for Cloud Storage*, The Huffington Post (2011).

[20] Quattor Community, *Quattor Toolkit*, http://quattor.org/.

[21] RESERVOIR Project, *RESERVOIR*, http://reservoir-project.eu/.

[22] Roy Thomas Fielding, *Architectural Styles and the Design of Network-based Software Architectures*, Ph.D. thesis, University of California, Irvine, 2000.

[23] SNIA, *Cloud Data Management Interface (CDMI)*, http://cdmi.sniacloud.com/.

[24] Storage Networking Industry Association (SNIA), http://www.snia.org/.

[25] StratusLab Collaboration, *StratusLab Marketplace*, Tech. report, StratusLab Project, 2 March 2011, http://stratuslab.eu/lib/exe/fetch.php/documents:marketplace-v3.0.pdf.

[26] StratusLab Project, *StratusLab website*, http://stratuslab.eu/.

[27] Telefónica Investigación y Desarrollo S.A. Unipersonal, *TCloud API Specification (Version 0.9)*, 26 January 2010, http://www.tid.es/files/doc/apis/TCloud_API_Spec_v0.9.pdf.